2

TASK: US40
CDRL: 03068
30 May 1991

US40 - Domain Specific Environment Repository Composite Paradigm Report

Informal Technical Data

STARS-SC-03068/001/00

30 May 1991

TASK: US40
CDRL: 03068
30 May 1991

INFORMAL TECHNICAL REPORT

For The

SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

Domain Specific Environment Repository
Composite Paradigm Report

STARS-SC-03068/001/00
Publication No. GR-7670-1195(NP)
30 May 1991

Data Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0003

Prepared for:

Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:

TRW
under contract to
Unisys Defense Systems, Inc.
Tactical Systems Division
12010 Sunrise Valley Drive
Reston, VA 22091

Distribution Statement "C"
per DoD Directive 5230.24
Distribution Authorized to U.S. Government Agencies and their Contractors:
Administrative (30 May 1991)

TASK: US40
CDRL: 03068
30 May 1991

Data ID: STARS-SC-03068/001/00

Distribution Statement "C"
per DoD Directive 5230.24
Distribution Authorized to U.S. Government Agencies and their Contractors:
Administrative (30 May 1991)

Developed by: TRW under contract to
Unisys Defense Systems, Inc.

This document, developed under the Software Technology for Adaptable, Reliable Systems (STARS) program, is approved for release under Distribution "C" of the Scientific and Technical Information Program Classification Scheme (DoD Directive 5230.24) unless otherwise indicated. Sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA) under contract F19628-88-D-0031, the STARS program is supported by the military services, SEI, and MITRE, with the U.S. Air Force as the executive contracting agent.

Permission to use, copy, modify, and comment on this document for purposes stated under Distribution "C" and without fee is hereby granted, provided that this notice appears in each whole or partial copy. This document retains Contractor indemnification to The Government regarding copyrights pursuant to the above referenced STARS contract. The Government disclaims all responsibility against liability, including costs and expenses for violation of proprietary rights, or copyrights arising out of the creation or use of this document.

In addition, the Government, Unisys, and its subcontractors disclaim all warranties with regard to this document, including all implied warranties of merchantability and fitness, and in no event shall the Government, Unisys, or its subcontractor(s) be liable for any special, indirect or consequential damages or any damages whatsoever resulting from the loss of use, data, or profits, whether in action of contract, negligence or other tortious action, arising in connection with the use or performance of this document.

TASK: US40
CDRL: 03068
30 May 1991

INFORMAL TECHNICAL REPORT
Domain Specific Environment Repository
Composite Paradigm Report

Approvals:

Richard E. Creps — 6/24/91

Task Manager *Richard E. Creps* — *Date*

(Signatures on File)

TASK: US40
CDRL: 03068
30 May 1991

INFORMAL TECHNICAL REPORT
For The
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

Domain Specific Environment Repository
Composite Paradigm Report

STARS-SC-03068/001/00
Publication No. GR-7670-1195(NP)
30 May 1991

Data Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0003

Prepared for:

Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:

TRW
under contract to
Unisys Defense Systems, Inc.
Tactical Systems Division
12010 Sunrise Valley Drive
Reston, VA 22091

Contents

List of Figures

1 INTRODUCTION

This document defines a risk-driven Composite Paradigm as an interim step toward the definition of a reuse-based paradigm for the development of trusted Navy Command and Control (C^2) systems. This report represents an adaptation and melding of previous DARPA, SEI, STARS and industry process modeling work, as appropriate, in the areas of risk-based activities; high performance, trusted system developments; software reuse; library support for reuse; and trusted system reuse issues. It directly addresses the STARS goals for a technology for building adaptable, highly reliable and cost effective software systems. In particular, it provides an initial framework for the development of trusted systems, a high risk domain growing in importance as the need for more complex systems and more systems interoperability increases.

This report presents a tutorial description of the Composite Paradigm for a reuse-driven system development. It does not provide specific guidance and explicit steps for application of the paradigm to an actual project. While outside the scope of the current subtask, documented prescriptive guidance for the paradigm application, for example, in a project manager's handbook, would facilitate its use and is a desirable future task.

This is the first of two major report deliverables for STARS Subtask US40.2. The final subtask report will provide a risk-driven, reasoning-based paradigm tailored to the trusted Navy C^2 systems domain. The final report will include a Draft Navy C^2 Risks and Characteristics Report as an appendix.

In this report, the words "paradigm" and "process model" are used interchangeably. The Composite Paradigm defined in Section 3 is denoted the STARS Composite Process Model (SCPM).

1.1 Background

The Phase I Process Model results of the DARPA/ISTO funded Advanced Computing Systems (ACS) Project at TRW provides a basis for the STARS Composite Process Model (SCPM). The information in this report is derived from descriptions of the DARPA ACS Project Process Model documented in [7] and the analysis of the role of reuse requirements in the development process. The development of systems requiring trust and high performance requires an increased, early emphasis on clear identification of risks, risk mitigation activities and development process controls. For a specific application, this emphasis includes the risks and characteristics native to the application domain. The domain aspects for tailoring to a Navy Command and Control system development will be addressed in the final task report. STARS planning includes work toward reuse processes, reuse libraries and domain specific environments with a goal of instantiation of a domain-specific Software Engineering Environment (SEE) for reuse. For this report, the SCPM activities are based on the assumption that the required domain analyses have been performed prior to the establishment of reuse requirements for a system to be developed under the paradigm. In addition, the existence of a STARS reuse infrastructure is a fundamental assumption for the SCPM descriptions. The final subtask report will describe domain analysis activities and the establishment of early reuse requirements.

The risk-driven characteristics of the SCPM are rooted in the Boehm Spiral Model [1]. Starting with the Spiral Model as a foundation, key elements of the DARPA ACS trusted system Process Model were identified. As described in [7], the key elements of the DARPA ACS Process Model are the following:

- The domination of the development process by risk management;
- The integration of engineering for trust and performance;
- The specialization for Ada across multiple activities of the lifecycle;
- The integration of other software engineering techniques (analysis/assurance and configuration control).

The DARPA ACS Process Model was defined to integrate security, broad trust and performance engineering with a modern risk-driven system development paradigm for Ada. The traditional waterfall development process has often been ineffective as a model for large scale, complex systems, particularly those with stringent trust and performance requirements. The DARPA ACS Process Model is intended to guide and support the project process to increase the productivity of the development team and the quality of the resulting system while reducing the inherent project risks for that particular domain.

1.2 Focus of the current work

This task addresses the inadequacy of current software development paradigms, especially for trusted systems, and focuses on the adaptation of a STARS-relevant process model based on previous work. In this task the following results are being integrated as appropriate:

- The current results of the ACS Process Model work applicable to a STARS reuse process paradigm
- The results from process model application efforts;
- The results of the SEI process model research and the relevant STARS prime contractor initiatives.

The DARPA ACS Process Model foundation for high performance trusted systems in Ada provides an opportunity for software improvement within the STARS environment. This subtask leverages the TRW DARPA/ISTO process model work and applies it to specific reuse and application domain considerations.

The risk–driven, Spiral Model basis provides a foundation for a high integrity, high performance system development process that focuses on reuse principles. Specific risk mitigation approaches such as modeling and prototyping may provide candidate reuse components for high risk software development. A general definition of the basic spirals of activity that includes reuse considerations provides a foundation for reusable, tailorable objectives and transitioning criteria within the paradigm.

Each key element of the DARPA ACS Process Model was analyzed with respect to the reuse paradigm and other process model work as required. Reuse analysis is integrated into all aspects of the SCPM foundation. Process control and well-defined transitioning criteria in high–risk, early spirals of activity remain a primary consideration within the SCPM. For any application of a spiral-based process model, it will be essential to carefully define the transition points and what is meant by criteria satisfaction very early before moving to the next spirals of activity.

1.3 Objectives of the STARS Composite Paradigm

There are four principal objectives of the SCPM. They represent the original objectives of the DARPA ACS Process Model coupled with the primary objectives of this subtask:

1. To improve the overall quality of delivered products and to increase productivity;
2. To improve the likelihood that delivered products will satisfy their reuse, performance and trust requirements;
3. To facilitate asset reuse and to provide a reuse–based paradigm for the STARS context;
4. To be automatable and supportable by a domain specific STARS SEE.

The SCPM aids in achieving these goals by providing an initial framework for process activities. In addition, the goal for automation is addressed by the listing of proposed activities within the major cycles of the SCPM. Further refinement of these activities to more prescriptive process steps (once the domain analyses are applied) will support the determination of reuse process building blocks and the specification of and experimentation with process model representation and enactment, in this and proposed follow-on STARS subtasks.

Risk reduction is a hallmark of the SCPM. Issues of reuse and of high–performance trusted systems typically stretch both the state of practice and the state of the art, and therefore pose high development risks. Software reuse remains an area of research that requires careful planning, major re–education and the support of tools and technology before it becomes an effective means for software development improvement. As with any new technology, the implementation of reuse within the life cycle will initially be a high risk endeavor and perhaps not cost effective for the first few systems built. To address these high risk areas, the SCPM provides for explicit risk analysis and mitigation activities early in each major spiral, as well as a risk review at the end of each spiral cycle.

Productivity increases arise from a combination of obvious and subtle sources. The use of Ada, in the hands of proficient developers, with good tools, can increase productivity. There is some promise that software reuse can decrease life cycle costs and improve reliability. More subtle is the effect of parallel activities at different levels of development. A skilled manager will be able to take advantage of the flexibility afforded by the process model to deploy resources in an advantageous manner.

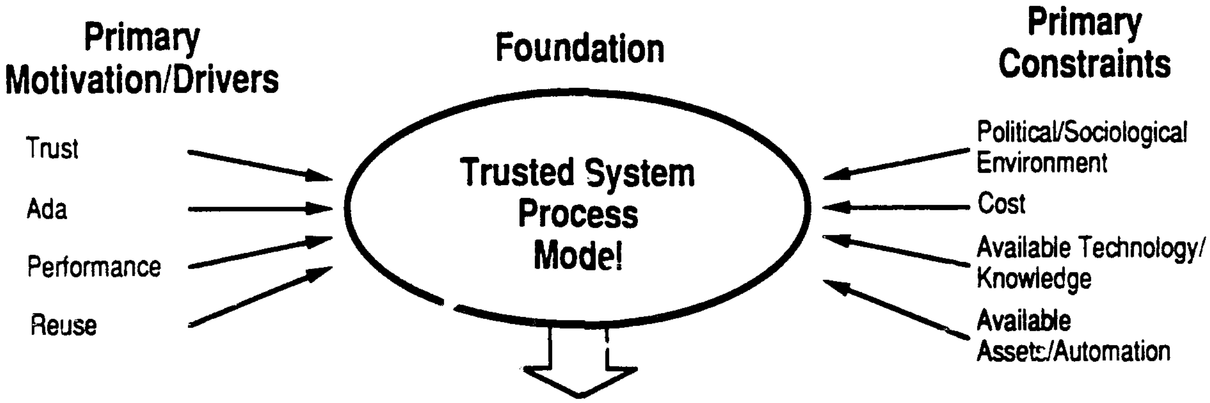
Early attention to risk analysis and resolution will have a salutary effect on quality, as will judicious use of a wide variety of reusable components and tools and formal and informal assurance techniques. The SCPM, with its focus on Ada use across multiple lifecycle activities, provides a framework that allows and encourages reuse of previously developed assets (e.g., designs and software), which may improve overall quality, and reduce costs. The flexibility of the SCPM will allow project management to use the most effective techniques suitable for producing a high-quality product.

1.4 Derivation of a STARS Composite Process Model (Paradigm)

The SCPM starts with the DARPA ACS Process Model as a foundation and enhances the model with an emphasis on reuse. Figure 1 illustrates the original motivations, drivers and key elements of the SCPM and the major considerations for reuse. Note: The italicized words illustrate the new areas for the SCPM. Detailed descriptions about the characteristics of the DARPA ACS Process Model can be found referenced in [7].

A reuse–driven process requirement implies the existence of fundamental capabilities to support the practice of software reuse. A basic assumption for the SCPM is that the domain specific basis for reuse, the asset library, and the support environment are in place. The reusable assets, their accompanying documentation, e.g., specifications, assurance and constraints information, to support all of the life cycle activities of the system development must be available and accessible. There must be an asset certification/qualification process defined and associated with reusable assets. For the development of trusted systems, there is a need for a high degree of confidence in the integrity of trusted assets. Management commitment and a clear understanding of the reuse process in general and for a specific domain are fundamental for reuse as a feasible process model driver. There must be an established library that supports the retrieval of reusable components and provides clear information to enable the reusers to reasonably assess the feasibility of specific and general reuse of components within their application domain. Foundations for the reuse libraries and the SEE are described in [17]. The subtask final report based on this SCPM foundation will include domain-based activities for reuse.

"Domain analysis is the first step in constructing reusable resources" [6] and the effectiveness of reuse within a particular application domain will not be known until actual systems are implemented using reusable assets. The success of a reuse library implementation will depend on its ability to support reuse activities in the development process lifecycle. The reuse library must reinforce the software reuse process activity as project personnel function within the software engineering environment. Tools and methodology must strongly guide and


Primary Motivation/Drivers
Foundation
Primary Constraints
Trust
Ada
Performance
Reuse
Trusted System Process Model
Political/Sociological Environment
Cost
Available Technology/ Knowledge
Available Assets/Automation

Key Elements of the STARS Composite Process Model

Risk Management

- Formal risk management techniques
- *Modeling*
- Planning for reuse
- Prototyping and demonstrations
- Analysis of reuse candidates
- Incremental development

Ada

- Homogeneous representation
- Language support for reuse
- Consistent metrics

Engineering for Trust, Performance and Reuse

- Architecture assessment (modeling, prototyping)
- Critical mechanisms prototyping
- Integration of critical reusable assets

Control and Assurance

- Reasoning-based analysis/assurance
- Reuse of assurance results
- Configuration management and control
- Control of reuse library

Figure 1: Derivation of the STARS Composite Process Model

assist the analysis, integration and evaluation processes for software reuse for requirements definition, design, development, testing and evaluation.

Under the SCPM, the system architect has responsibility and authority for trading reuse, trust and functionality in order to meet cost and schedule constraints. In addition, all developers and managers should be encouraged to receive training in reuse methodology, trust, and performance concepts and their interactions. Where specialists in reuse, trust or performance engineering are needed, they are merged into the development teams and share responsibility for product delivery, reusability, and implementation of reuse, trust and performance objectives.

The SCPM facilitates the analysis of available assets for reuse as a risk management technique at all levels of design and implementation. Ada's packaging and generic mechanisms are a deliberate attempt to assist such reuse. It is to be expected that reusable assets, used within their intended realm of applicability, would be more mature, demonstrating known and previously validated properties as compared to new components.

The impact of reusable software can be assessed at a number of points in the software lifecycle, starting with the software architecture definition and continuing through detailed design. Assuming the existence of reuse libraries, representing significant system functionality within a SEE, reuse should increase software quality because reused components will tend to be subject to extensive analysis and upgrading over a long period of time. Reuse of trusted functionality (designs, components, code) presents a significant challenge requiring additional research. Key issues in reuse of trusted components are explored in [5].

1.5 Relevant Documents

The following documents provided information that was useful either directly or indirectly for this report: [4, 17, 5, 3, 6, 8, 9, 10, 12, 16, 2, 11, 18]. They represent efforts that are relevant to the goals of a STARS–oriented process model for reuse and trust within specified application domains.

2 THE STARS COMPOSITE PROCESS MODEL

The SCPM for reuse–driven, high–performance trusted systems in Ada addresses its objectives through two primary strategies. First, the SCPM stresses the early identification of risks, and organizes subsequent development activities to mitigate them. Second, the SCPM calls for integration of reuse, trust and performance engineering with modern software engineering practices. The reuse process activities are based on an assumption for this report that the fundamental domain drivers and characteristics are well understood and established for a particular system development. In actual practice, preliminary domain analysis activities, generic architecture definitions and reusable asset analysis and management processes will be performed prior to the initiation of a reuse–based development project. These activities will be addressed in the subtask final report.

The SCPM emphasizes the integration of various engineering practices, the use of Ada throughout multiple phases of development, and the inclusion of a spectrum of risk–reducing development, analysis, and reasoning–based assurance techniques and tools. Moreover, because of the SCPM dynamic activity sequencing and reuse emphasis, the importance of configuration management as a mechanism for coordination and status accounting is heightened. In planning each major stage of project activity, transitioning criteria and a means for closure must be well established to reduce the uncertainty of project management and ensure reasonable progress.

2.1 Conceptual View

Figure 2 provides a conceptual representation of the original SCPM foundation, the DARPA ACS enhanced and tailored Spiral Model for trusted system development and illustrates its possible spirals, activities and milestones. As in [1], the software development process over time is depicted as a clockwise spiral that begins in the middle of a system of polar coordinates. The radial coordinate (distance from the origin) represents cumulative project cost; the angular coordinate represents progress with respect to the objectives of a specific 360 degree cycle. During each cycle, four generic classes of activities are carried out in sequence. Each class is represented as an activity quadrant; these are traversed clockwise during each cycle. Elapsed time is portrayed as an elastic, unitless dimension in which the amount of time spent in each quadrant varies from spiral to spiral.

In the first quadrant (beginning at "9 o'clock") objectives, alternatives for achieving those objectives, and constraints on possible alternatives are identified. In the second quadrant, alternatives are evaluated in terms of probability and cost of failure, and potential magnitude of payoff. This is primarily a task of information collection and analysis, involving prototyping, analytic modeling, interviews and surveys, literature searches andor other techniques. In the third quadrant, one or more of the favorable alternatives are selected and pursued. In the early spirals, "pursuit" may simply mean making and documenting strategic technical decisions, e.g., selection of a hardware vendor, choice of distributed versus centralized architecture, etc. In later spirals, it may mean further refinement of prototypes, formal analysis and modeling, or undertaking such "product development" steps as producing technical plans and specifications, designs, or a completed operational system. Reasoning–based techniques have a role in both the second and third quadrants as the attendant modeling, specification and analysis activities can either support the risk mitigation process by helping to select a viable alternative, or by producing a product such as a formal specification or system performance model that becomes part of the system design and, ultimately, part of the delivered system. In the fourth quadrant, progress against objectives is assessed; recommendations and plans for continuing, changing, or terminating the activity are prepared. The fourth quadrant concludes with a commitment to the plan (followed by initiation of the next spiral), or with termination of the process. To the extent that the fourth quadrant of a cycle is dominated by planning rather than progress assessment, the fourth quadrant activities may be more closely related to the objectives of the next cycle than the cycle of which they are nominally part.

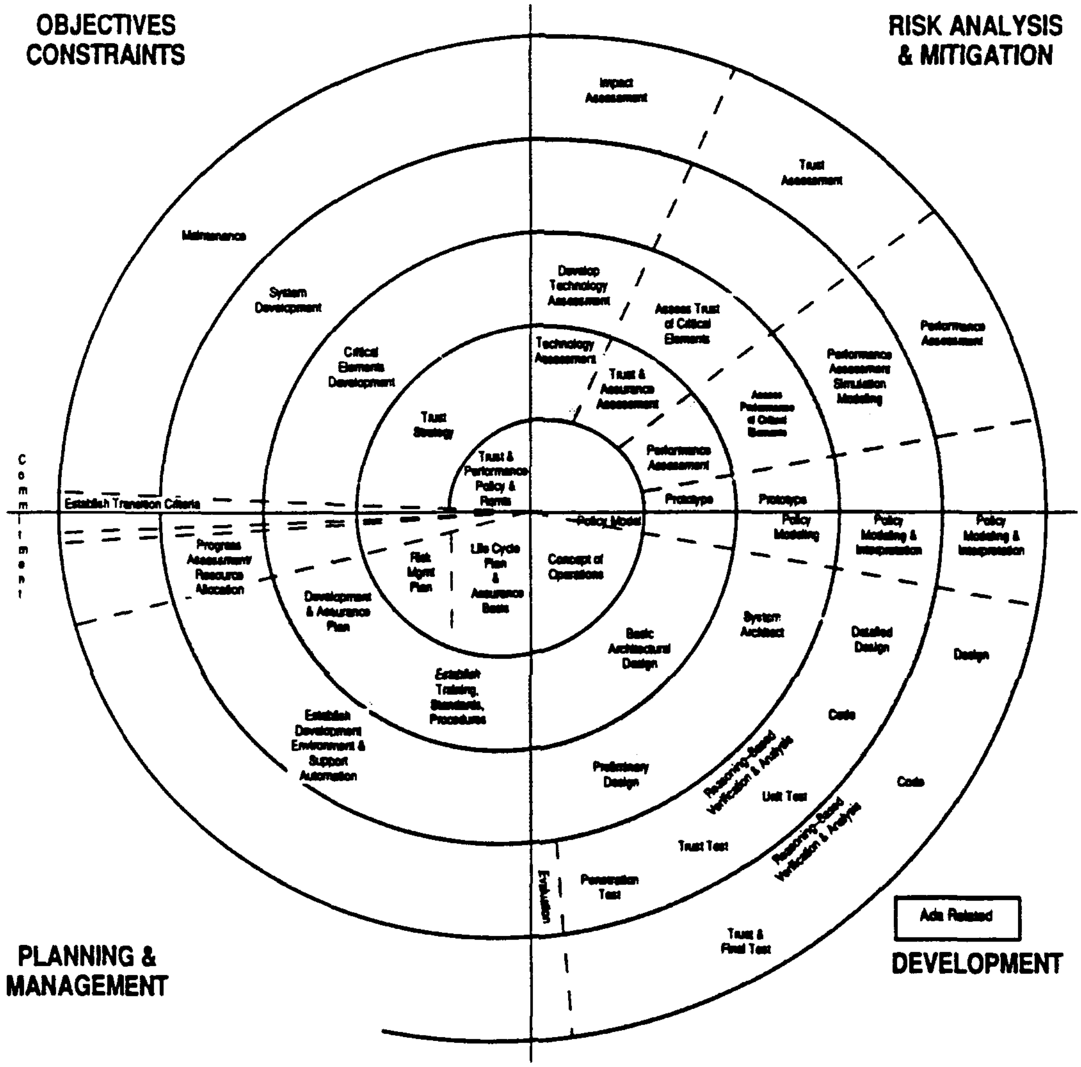

OBJECTIVES
CONSTRAINTS
RISK ANALYSIS
& MITIGATION
Impact Assessment
Trust Assessment
Maintenance
System Development
Develop Technology Assessment
Assess Trust of Critical Elements
Performance Assessment
Critical Elements Development
Technology Assessment
Trust & Assurance Assessment
Assess Performance of Critical Elements
Performance Assessment Simulation Modeling
Trust Strategy
Trust & Performance Policy & Rqmts
Performance Assessment
Commitment
Establish Transition Criteria
Prototype
Prototype
Policy Model
Policy Modeling
Policy Modeling & Interpretation
Policy Modeling & Interpretation
Progress Assessment/ Resource Allocation
Risk Mgmt Plan
Life Cycle Plan & Assurance Basis
Concept of Operations
Development & Assurance Plan
System Architect
Detailed Design
Design
Basic Architectural Design
Establish Training, Standards, Procedures
Code
Establish Development Environment & Support Automation
Preliminary Design
Reasoning-Based Verification & Analysis
Unit Test
Code
Trust Test
Reasoning-Based Verification & Analysis
Evaluation
Penetration Test
Ada Related
Trust & Final Test
PLANNING &
MANAGEMENT
DEVELOPMENT

Figure 2: Initial Foundation for the STARS Composite Paradigm

While the Boehm Spiral Model defines the activities to be carried out in each quadrant in very generic terms, the DARPA ACS Process Model suggests that particular patterns of activities within certain quadrants may be desirable and provides a more detailed specific elaboration of activities. This elaboration is based on significant experience and lessons learned in the development of mission-critical and commercially viable trusted systems. These activity patterns stem from iterative attempts over several cycles to incrementally reduce crucial risks common to high-performance trusted systems in Ada. This is approached by the partitioning of the second and third quadrants into more specialized activity sectors. Most spiral cycles will include activities belonging to one or more of these sectors. In addition, activities will be more prescribed and specific within a particular domain of interest. In our adaptation for the SCPM, this will involve the following general activities:

- Technology assessment and reassessment
- Reuse assessment
- Trust and assurance assessment
- Performancification assessment
- Asset qualification assessment;
- Prototyping
- Policy modeling

Assuming a domain specific basis, the SCPM incorporates the considerations for reuse and proposes reuse activities within each spiral and within the quadrant segments. Figures 3 - 7 collectively present a conceptual view of the SCPM. In the very early stages of reuse technology, it will be appropriate to define an initial spiral for the actual domain analysis process. This will be analyzed in the domain specific work for subtask US40.2 and addressed in the final report.

As shown in the figures, early spiral cycles concentrate on reducing risk associated with reuse, trust policy, performance requirements, and address these risks with trust strategy and prototyping or accelerated development of critical system components. Later activities are characterized by a tailoring of engineering standards, practices, analyses, and tools in order to integrate reuse, trust, performance, and Ada, into traditional software and system engineering. System development then proceeds, guided by the tailored development process.

Although it is the intent of the SCPM to have resolved the major development risks after a few early spiral cycles, *in practice* risks will persist after this period. As a result, supplementary risk-reducing analysis, tools, and techniques are recommended throughout the lifecycle. The final development, system operation and maintenance are illustrated using additional spirals. In practice, these spirals may represent multiple spirals of activity or may be performed in a more traditional phased set of activities depending on the degree of risk that still remains.

System development under the proposed SCPM differs from the traditional waterfall development model in that it may consist of loosely synchronized concurrent threads of development. In particular, the threads may be organized to follow an incremental development approach [2]. Moreover, each development thread may follow a non-traditional sequence of development activities. Management of the overall process requires a clear and early definition of the transition points that drive the decision to move from one spiral of activity to the next. The flexibility of the spiral process allows for changes in spiral goals and transitioning criteria as feasible for a specific project. However, this flexibility must be carefully managed and should be based on a pre-established control process (e.g., team consensus with management approval, etc.).

Because modification of a trusted system has the potential for invalidating the evaluation rating or certification of the system, maintenance phase activities are inherently risky. In addition, the assessment of assets for reuse qualification, reuse approach and prototyping activities may be invalidated by component modification As a result, the SCPM treats maintenance as requiring risk-management techniques similar to those employed in earlier life-cycle phases. This view is reflected in Figure 7 by the representation of maintenance as another spiral cycle.

2.2 Early Risk Spirals for Reuse, Trust and Performance

Although development risks will vary from system to system, some common risks are inherent within a particular application domain, in reuse-driven developments and in systems requiring high trust and performance. The SEI reuse lifecycle defined in [6] described a minimum set of activities within the life cycle and methodology phase of their four-phased model. These activities address the need to embed reuse into the daily development effort. The common patterns for crucial risk mitigation with respect to reuse, trust and high performance system development typically addressed in the early spirals of the SCPM are described in the sections below.

2.2.1 Initial Project Plans and Analysis of Reuse, Trust and Performance Requirements

The first major spiral of activity is dedicated to project planning and understanding the requirements. Initial system requirements for reuse and trust, including requirements for reuse approach, trust policy, assurances, asset qualification and trust evaluation, may be conceptually difficult, ambiguously stated, unrealistic, and in conflict with other requirements. In particular trust and performance requirements may be opposing, and the issues of reuse are further complicated by this conflict. Consequently, the spiral activities advanced by the SCPM include analysis of the cost, implications, and achievability of initial reuse, trust and performance requirements. The objective is to understand the statement of the problem with respect to reuse, trust and performance, and identify the flaws and obstacles. The analysis may include the following activities :

- Identification of reuse policy and goals;
- Clarification of trust policy;
- Review of trust principles and their historical interpretation and application;
- Initial assessment of trusted and untrusted reusable assets (other than Commercial-Off-The-Shelf (COTS) products) and their component level and system level reuse implications;
- Assessment of emerging trusted and trust-compatible COTS products, including discussions with vendors about plans for future products;
- Assessment of support capabilities of library and SEE and available technology for reuse and trust goals;
- Dialogue with evaluation and accreditation authorities to clarify trust criteria and evaluation procedures and implications of reuse of trusted assets;
- Identification of unachievable or high-risk trust and performance requirements;
- Development of written interpretations of reuse, trust and performance requirements, potentially useful to support a draft Concept of Operations document;
- Development of a life-cycle plan that emphasizes approximate budgetary and schedule milestones, incorporates reuse and risk management strategies, defines initial spiral transitionary criteria and describes the techniques and tools used to assess progress and to provide management visibility and control during subsequent spirals;
- Development of a reuse plan (for current reuse and future reuse capabilities);
- Clarification of basis for assurance of trust policy enforcement in developing systems, particularly for reusable, trusted assets;
- Initial identification and analysis of major project risks;
- Development of a risk management plan and establishment of transitioning criteria for next project spiral.

These activities are illustrated in Figure 3 which presents a conceptual view of Spiral 1. In actual practice, some of these activities may be combined, may not be required or may be addressed in later spirals depending on project size and complexity and specific requirements.

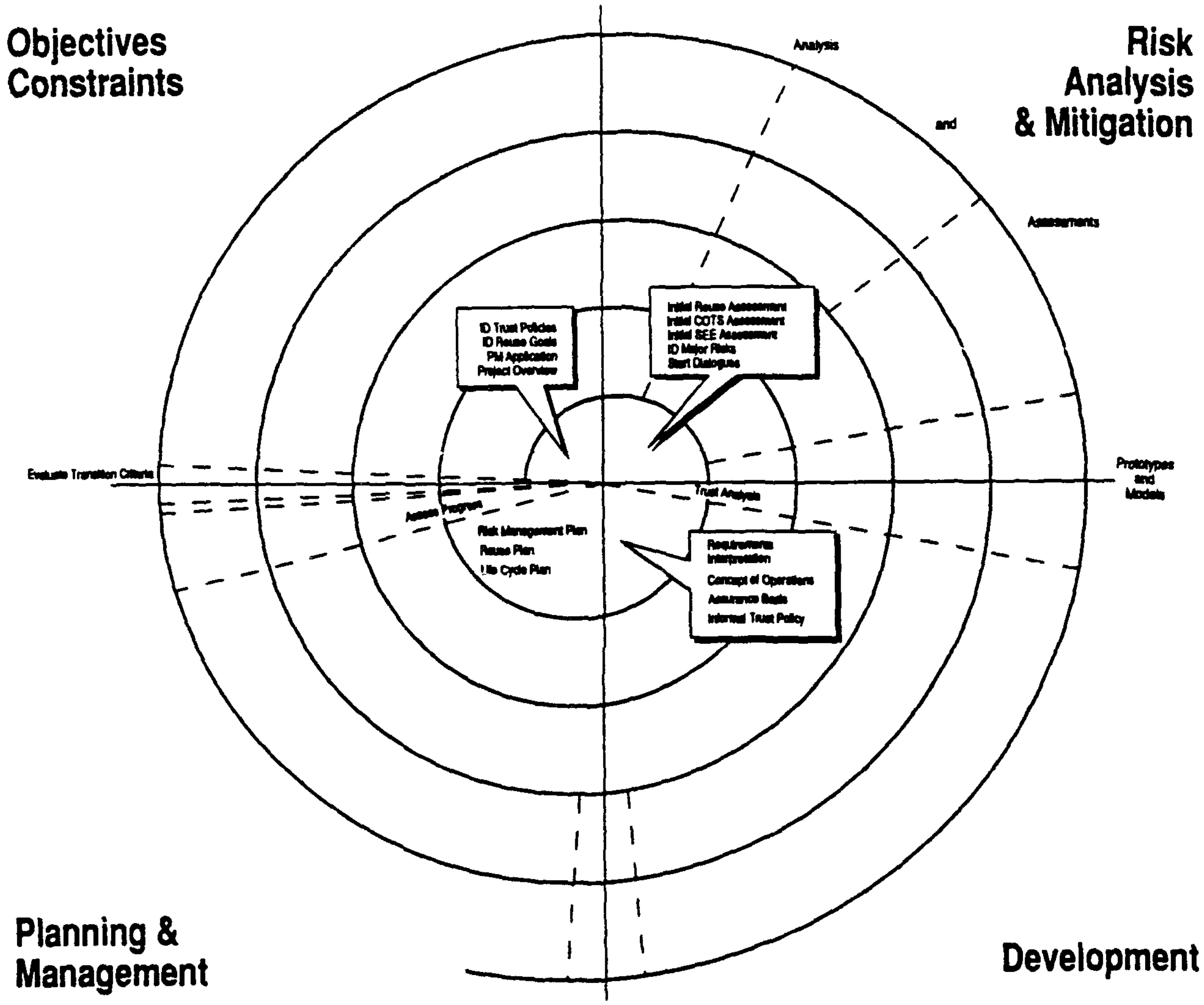


Figure 3: A Conceptual View of Spiral 1: Initial Project Plans and Requirements Analysis

2.2.2 Reuse and Trust Enforcement Strategy and Basic Architecture

After the initial reuse and trust requirements analysis, a strategy for enforcing the reuse methodology and the trust policy must be developed. Additional assessments may be appropriate for technology considerations, process model application and SEE support including reuse library mechanisms, automated process management, and risk management, asset qualifier/tracker and language analysis tools.

The trust policy refinement is perhaps best accomplished by formulating a hypothetical trust enforcement architecture that embodies high-risk trust features and requirements and incorporates trusted, reusable assets as feasible. The hypothetical architecture is then evaluated for expected performance, robustness, functionality, and impact on untrusted component behavior and structure. The components of the hypothetical architecture may include existing hardware or software components that have been adapted for trust, emerging trusted COTS products, or entirely new custom-developed elements. The evaluation of the hypothetical architecture may be limited to 'paper and pencil' analysis, or may involve hands-on experiments or prototypes to investigate key characteristics of potential components.

The use of formal methods to model and analyze the required trust and performance properties of the architecture may also be appropriate. An assurance plan is needed to define the appropriate assurance activities based on earlier assessments of reused components, trust needs and cost feasibility. Unachievable trust and performance requirements, and high-risk architectural decisions are identified. Interpretations of trust evaluation criteria that are non-trivial, or novel in approach, are outlined, the impacts of reuse are identified, and the rationale may require discussion with evaluation or accreditation authorities. Initial performance budgets for key trust features may also be identified. Training standards and procedures for employees and future system users that emphasize reuse and trust principles must be developed.

The project schedule as well as the risk management and reuse plans may need revision. The plans must consider such reuse and trust issues as reevaluation of trusted components, reuse and integration of trusted assets in a new environment and integration of heterogeneous trusted components. These plans establish the risk mitigation activities and transitioning criteria for the next project spiral(s). A project assessment is necessary before transitioning to the next project spiral.

The activities described above are illustrated in Figure 4, A Conceptual View of Spiral 2. In summary they may include:

- Development/refinement of reuse strategy;
- Refinement of trust strategy/philosophy (reuse and trusted computing base (TCB) constraints, etc.);
- Additional assessments of technology as needed;

- Objectives determination and assessment and tracking of early process model application;
- Assessment of initial SEE support;
- Initiation, as required, of prototypes to validate/refine trust and reuse approaches;
- Development of trust policy model (formal or informal as required);
- Basic architecture definition that provides required trust and applies reuse as feasible;
- Tailoring SEE for project–specific needs;
- Conducting reviews as required;
- Documenting engineering notes;
- Establishment of training standards and procedures;
- Revisitation and update of project schedule and lifecycle plan;
- Development of assurance plan;
- Assessment of project progress and transitioning criteria achievement;
- Revision of the reuse and risk management plans as necessary and establishment of transitioning criteria for the next project spiral.

2.2.3 Critical Elements and Architecture Refinement

This set of risk–reduction activities verifies the achievability of reuse, trust and performance requirements, and establishes a foundation for system design. This is accomplished by prototyping critical elements of a candidate policy enforcement architecture and/or experimenting with critical reusable assets. These activities are to provide empirical evidence that an architectural solution is within reach and to define its underlying approach. The prototype may be based on a generic reuse architecture for a domain with reusable assets or built from real components, stubs, or a combination of the two. Ada may be used even at this early stage. The hypothetical architecture must show evidence of:

- Successfully applying and integrating reusable assets;
- Enforcing reuse methodology/designing for future reuse;
- Satisfying trust performance requirements and not preventing the satisfaction of other performance requirements;
- Enforcing trust policy; and

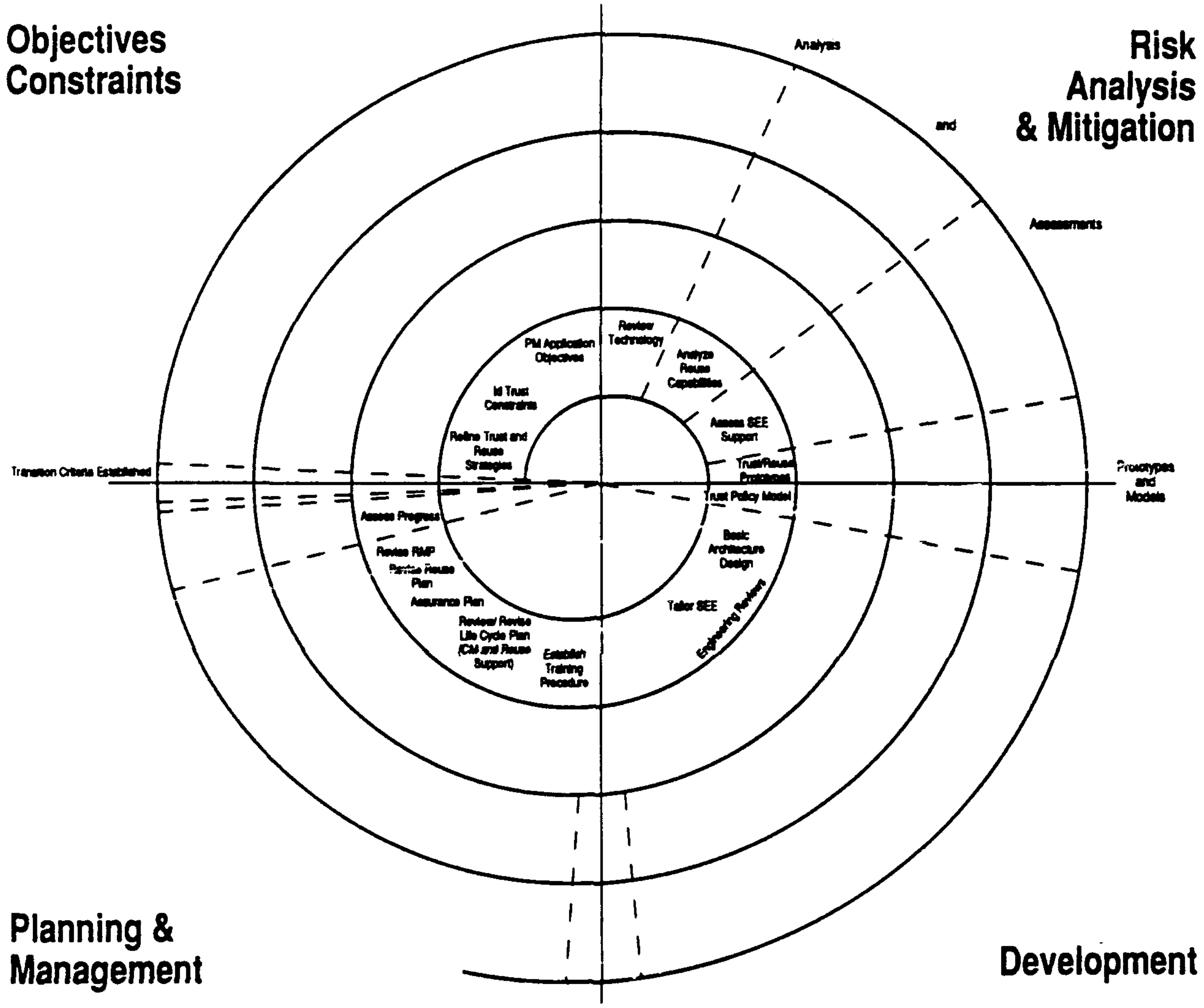


Figure 4: A Conceptual View of Spiral 2: Reuse and Trust Strategy and Basic Architecture

- Complying with trust assurance requirements, primarily well–structuredness.

The prototype evaluations may also assess the impact of the architecture's external interface on reusability and on both untrusted components and human users. An inability to hypothesize a satisfactory architecture may indicate that more drastic risk mitigation measures should be considered, such as the relaxation of reuse, trust or performance requirements, cost, or schedule. Depending upon the sophistication and success of the prototype and the scale of other risks, the prototype may be a throw–away that simply verifies the feasibility of requirements, or it may become the base from which the system's architecture evolves and/or may consist of reusable assets that can be applied to future system developments.

The spiral activities that may occur during preliminary design are illustrated in Figure 5, a Conceptual View of Spiral 3. They are:

- Incorporation of TCB and reuse constraints into critical element considerations;
- Planning critical element prototypes and experiments;
- Development of critical elements;
- Experimental integration of new and reusable critical elements;
- Assessment of process model application;
- Performance assessment of critical components;
- Assessment of prototype reuse qualification;
- Policy modeling;
- Reassessment of risks;
- Prototyping trust and reuse approaches;
- Establishing system architecture;
- Documenting preliminary design·
- Compiling, documenting design assurance evidence;
- Conducting reviews as needed;
- Documenting engineering notes;
- Assessment of progress and allocation of resources;
- Revision of project schedule;
- Revision of risk management plan.

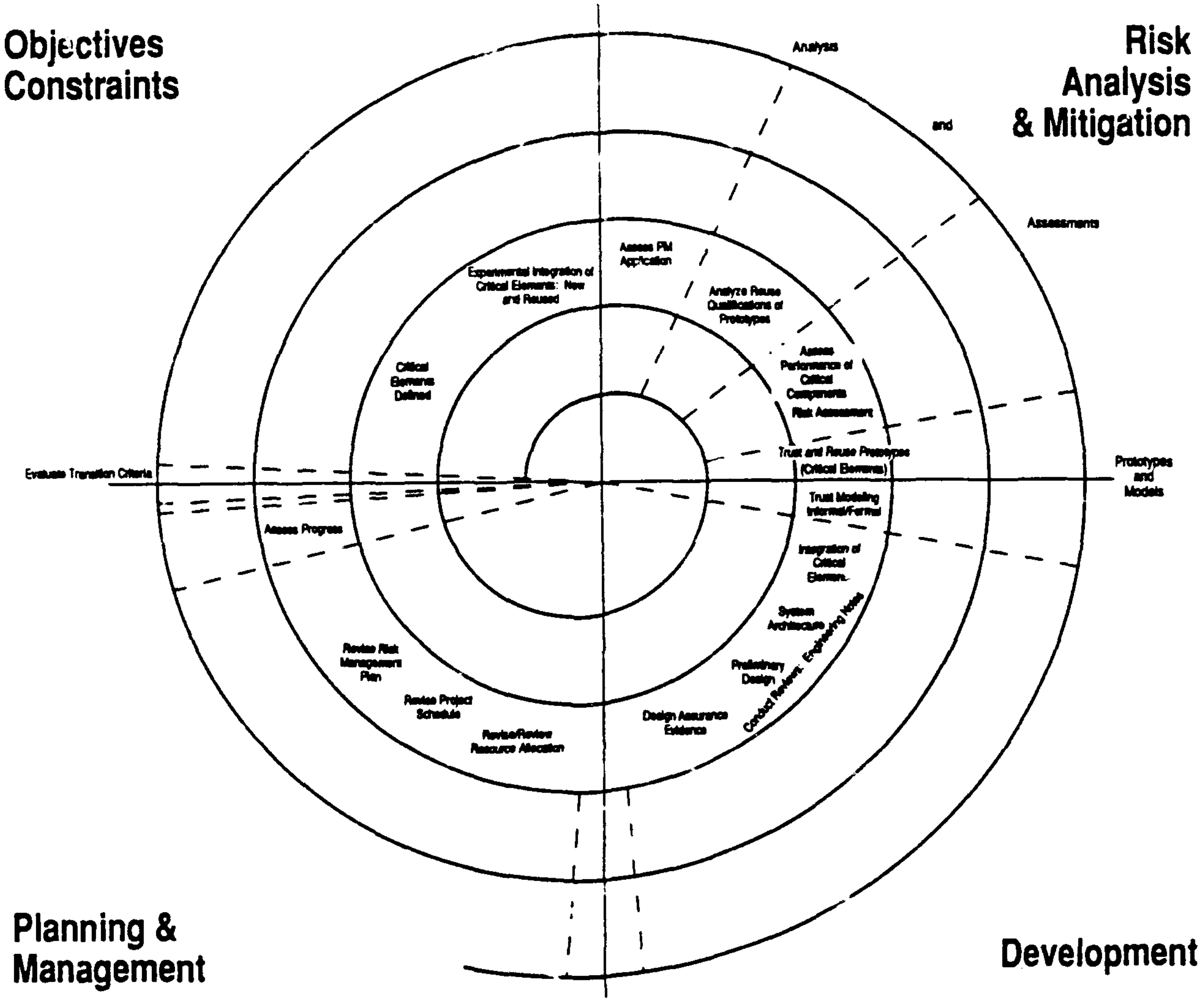

Objectives
Constraints
Risk
Analysis
& Mitigation
Analysis
and
Assessments
Planning &
Management
Development
Evaluate Transition Criteria
Prototypes
and
Models
Experimental Integration of Critical Elements: New and Reused
Assess PM Application
Analyze Reuse Qualifications of Prototypes
Assess Performance of Critical Components
Risk Assessment
Trust and Reuse Prototypes (Critical Elements)
Trust Modeling Informal/Formal
Integration of Critical Elements
System Architecture
Preliminary Design
Conduct Reviews: Engineering Notes
Design Assurance Evidence
Critical Elements Defined
Assess Progress
Revise Risk Management Plan
Revise Project Schedule
Revise/Review Resource Allocation

Figure 5: A Conceptual View of Spiral 3: Critical Elements and Architecture

The activities performed during early design and the number of spirals required will vary according to the needs of a particular project. In particular, once reuse technology is well established for a particular application domain, the preliminary design activities may be simplified enough to require mainly reuse analysis. If a sound basis for trust is well established, integration of software assets will potentially enable integrators and developers to reuse heterogeneous components and still preserve trust for a particular environment.

2.3 Development of a Trust Engineering Process That Incorporates Reuse

To develop an integrated engineering process, an analysis of the commonality of reuse engineering, trust engineering and traditional system/software engineering, especially as related to performance, is undertaken. This analysis examines assets, products, documentation, activities, skills, and responsibilities to determine how integration of engineering concerns should best be accomplished. In addition, a technology assessment of reuse capabilities and of trust-supportive tools within the SEE is carried out.

If formal verification technology is required, the choice of a formal specification language and tool set is carefully weighed, as it is likely to rigidly constrain design options for policy enforcing components. The chosen verification methodology and tools must be integrated into project planning and will involve SEE tailoring for the formal, reasoning-based engineering support. In addition, a verification plan for timely coordination and feedback between designers and design verifiers is developed to reduce the risk that a verification obstacle will force major last-minute design changes. These analyses may produce such by-products as trust training plans, outlines for tailored documentation, high-level requirements for automated tools and environments, and software standards and practices manuals. The reuse of trusted assets will require strong confidence in asset integrity as well as early agreements with certification/accreditation officials on the role and acceptability of reuse in the verification process.

For systems that are trusted with respect to Trusted Computer System Evaluation Criteria (TCSEC) confidentiality, trust assurance criteria are predetermined although the interpretation for a specific system development still needs to be done. For systems trusted with respect to other policies such as integrity, safety, or denial of service, the trust assurance criteria must first be refined as part of defining an integrated engineering process. For some of these policies, e.g., nuclear weapons safety, a large body of assurance analysis techniques exists, while for others the choice may be more limited.

If formal methods are required to assure system trust under a reuse methodology, reuse and trust analyses are complicated by such issues as the applicability of the previous formal methods of a trusted asset to a new environment and the need for partial or total reverification of a reused trusted asset. Trusted assets that are candidates for reuse include software components, trust specifications, trust models, trust policy statements and documented verification proofs and informal correspondences Trust assets are currently defined in terms of an overall system or product and not in terms of composable pieces. As pointed out in [5], composability of trusted systems is currently an open research issue in computer security.

The document lists the following aspects of the topic as controversial: distinguishing between trust policy composability, model composability and specification composability; and defining methods for composing policies, policy models and specifications. In addition, [5] discusses the need for approaches to composable specification and verification systems and elaborates on composable verification, testing and covert channel analysis issues.

2.4 Later Spirals

The later spirals of activity involve product development for building reuse–driven, high–performance trusted systems and operations and maintenance of fielded systems. There is generally less emphasis on resolving major risks during development than in the earlier spirals. However, the maintenance activities may mirror some of the earlier spirals' risk mitigation since changes may introduce major new risks.

2.4.1 System Development and Assurance

The early spirals of the Model deal with resolving major risks in the feasibility, requirements, scope, and reuse and conceptual approach to building a high-performance trusted system, while the later activities are concerned with product building. Reuse planning and methodology strongly influence the development of new products both from a current use standpoint and the goals for future reuse. Reusable components may be shown to be consistent with a new or reused specification in a new environment and/or with respect to new interfaces. Approximations used in performance models may be validated as actual components become available. The SCPM proposed here differs from the traditional waterfall model in the following ways:

- The SCPM recognizes the continuing need for risk–assessment and risk–mitigation activities (including reasoning–based analysis, modeling and prototyping), and explicitly calls for their presence throughout major portions of the development process. In addition, to the extent possible, software development techniques and tools as well as reuse support are incorporated in the SEE to further reduce risks.
- The SCPM allows concurrent threads of development activities that may traverse the traditional progression of software product–phases in loosely synchronized manner.
- The SCPM allows each thread to follow non–traditional progressions of activities, such as those advanced by the evolutionary development model [14], or the transform development model [13], if appropriate to the domain.

Figure 6, A Conceptual View of Spiral 4 illustrates the following possible activities during the development and assurance stages of the process:

- Development of the system (consists of many subactivities; may be incremental);

- Reuse of acceptable assets within the system development;
- Monitoring the application of the process model;
- Assessing maintenance and reuse requirements
- Evaluation and certification of components;
- Assessment of asset qualifications;
- Assessment of component and system performance;
- Analyses and assessments of other issues as required;
- Interpretation/proving the policy model(s);
- Additional/continuing prototyping as required;
- Coding;
- Documentation of detailed design;
- Documentation for system users;
- Documentation of other assurance evidence as required;
- Testing and evaluation for all requirements including reuse, trust, penetration, performance and system-wide testing;
- Documentation of maintainability and evolvability including trust, performance and reuse issues;
- Performing reviews and walkthroughs;
- Documenting engineering notes;
- Documentation of assets for future reuse;
- Accreditation of system;
- Planning for operations and maintenance;
- Tracking CM (reuse and trust);
- Developing guidelines for maintenance and reuse;
- Reviewing lessons learned;
- Revision of the RMP.

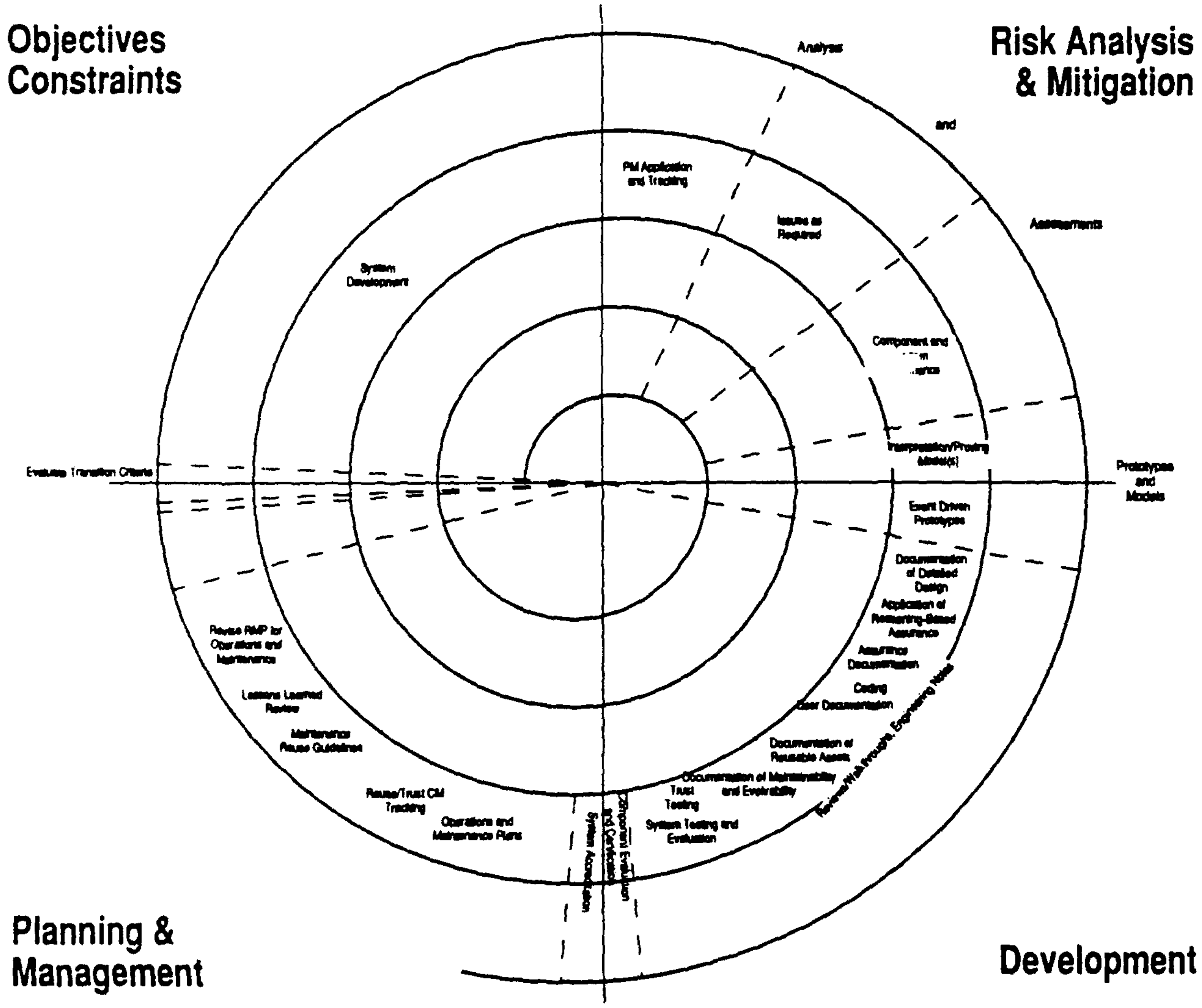


Figure 6: A Conceptual View of Spiral 4: System Development and Assurance (May Be Incremental Over Multiple Spirals)

Although the development and assurance activities are conceptualized as occurring in a fourth spiral, the required activities may occur over multiple spirals depending on the degree and number of project risks that occur or remain during system development. The required set of activities for a particular project development could be conducted within a phase-oriented pıocess such as the standard waterfall paradigm if the development risks have been reduced to a very low level. Multiple, concurrent or phased spirals may also be used to represent incremental stages of coding and testing that may be separate or may depend on other spirals.

2.4.2 Maintenance

For most high-performance trusted systems, maintenance is the phase that dominates the lifecycle costs. It has traditionally introduced risks, particularly those associated with system degradation caused by modifications that over time diminish the integrity and clarity of the system design. Attempting to control maintenance costs and activities has been the significant driver for much of software engineering research and development.

The advance of a successful reuse technology should greatly reduce the traditional problems associated with software maintenance. Engineering for reuse is analogous to engineering for ease of maintenance. The desirable characteristics of reusable assets are much the same as those of maintainable assets. The availability of reusable assets and the associated information within a SEE containing a knowledge-based reuse library will provide strong support for maintenance engineering.

Use of the SCPM during maintenance follows the same pattern that was applied during development. Objectives, alternatives, and constraints are examined. Risks associated with the candidate modifications are assessed for reuse, trust and performance implications, and an approach with minimal impact is selected. At this point, the use of formal models and specifications developed during the system construction may provide a method for evaluating the impact of proposed changes without the trial and error process that often accompanies maintenance efforts.

Maintenance modifications are achieved by updating all of the relevant development documents. Strict configuration management of the products is required for both reuse and trust. The implications of modifications should be well documented to support reuse qualification and to facilitate re–evaluation, if required. Maintenance activity, with modifications collected or grouped so the result is a new version of the system, represents additional spirals in the SCPM. Reuse issues may involve the qualification of both the old and new asset versions and the provision of rationale for maintaining both in a reuse library. Reuse qualification/certification methodology must apply to maintenance of all assets, and the control of asset versions with rationale for maintaining older versions is a critical requirement for reuse.

Figure 7, A Conceptual View of Spiral 5, illustrates the possible activities within the quadrants and sectors of a maintenance spiral for systems requiring trust and reuse. The maintenance spiral(s) may involve:

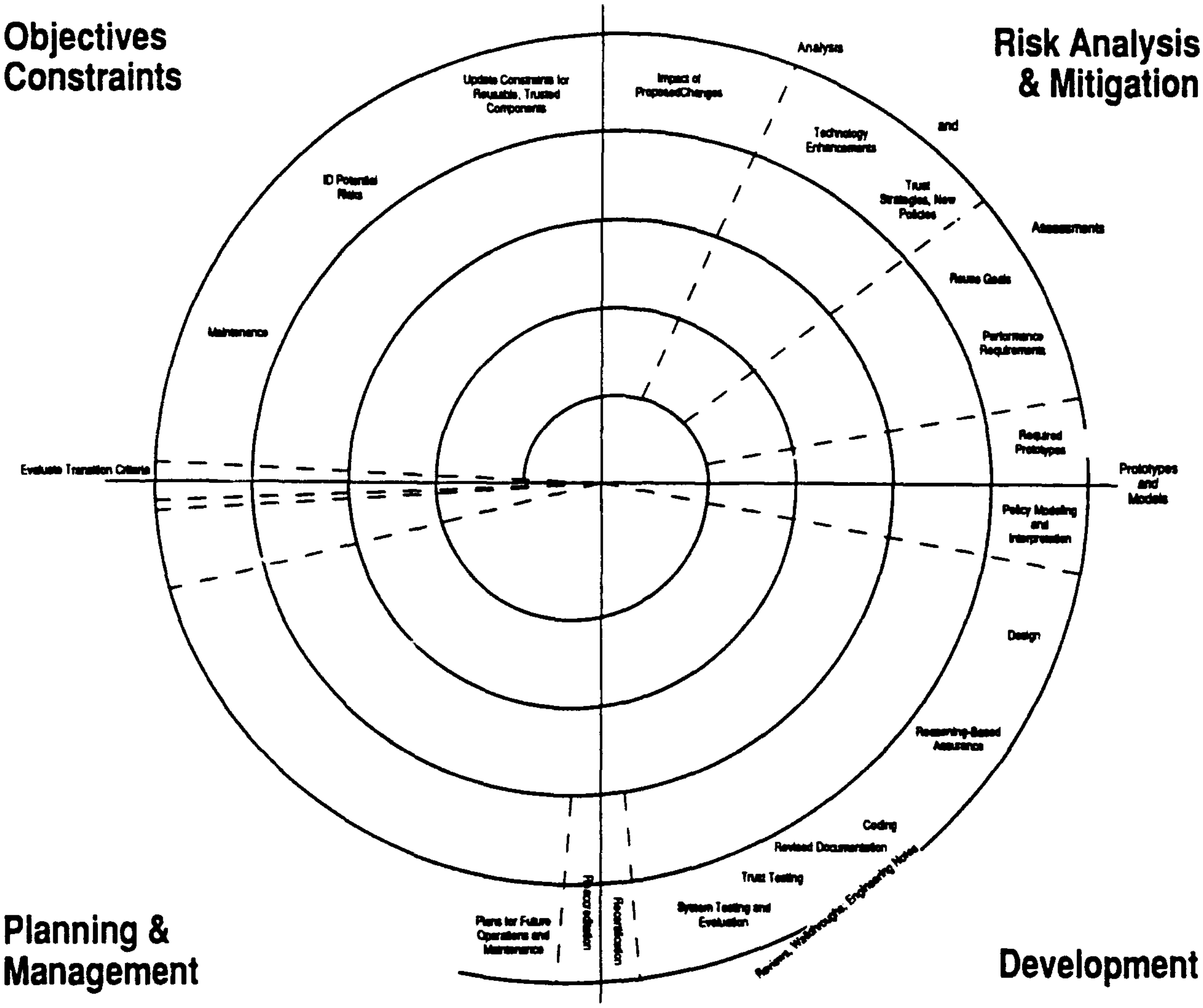

Objectives Constraints
Risk Analysis & Mitigation
Planning & Management
Development
Update Constraints for Reusable, Trusted Components
Impact of ProposedChanges
Analysis
and
Assessments
Technology Enhancements
Trust Strategies, New Policies
Reuse Goals
Performance Requirements
Required Prototypes
Prototypes and Models
Policy Modeling and Interpretation
Design
Reasoning-Based Assurance
Coding
Revised Documentation
Trust Testing
System Testing and Evaluation
Reviews, Walkthroughs, Engineering Notes
Recertification
Reaccreditation
Plans for Future Operations and Maintenance
Evaluate Transition Criteria
Maintenance
ID Potential Risks

Figure 7: A Conceptual View of Spiral 5: Maintenance

- Implementation of mechanisms for tracking of changes to reusable assets;
- Implementation of mechanisms for tracking and analysis of changes to trusted elements;
- Maintenance of baselined assets;
- Identification of potential maintenance risks;
- Updating constraints for trust and reuse;
- Reuse, trust and performance impact assessment of proposed changes;

- Assessment of technology to support reuse and trust maintenance;
- Assessment of asset qualification after modifications;
- Identification of major new risks introduced by change and risk mitigation activities;
- Modeling and interpretation of trust policy for continued adherence or possible revisions as required;
- Development of prototypes as required;
- Development of design/design revisions (software, hardware and documentation) as required;
- Application of reasoning-based analysis and verification as required;
- Performing reviews and walkthroughs as required;
- Documenting engineering notes;
- Retesting/testing and evaluation for reuse, trust, and performance as required;
- Reevaluation and recertification of elements as required;
- Reaccreditation of system trust as required;
- Revision of risk management plan as required;
- Reviewing lessons learned;
- Planning for future operations and maintenance.

In practice, the maintenance spiral could be partitioned into a number of spirals that address the specific risks associated with system changes. Depending on the amount of effort involved and the degree of risk, the spirals may be similar to those used to address design and development risks in the initial system development.

Maintenance for trusted systems is a challenging task since modification to the trusted portion of the system has the potential for invalidating the evaluation rating/certification of components and/or the accreditation of the system. For nuclear systems, certifications are achieved via arduous analysis and examination of the trusted system and its documentation by teams of certifiers and analysts who scrutinize critical components of the system, its design, code, analytic results, and test results. Since implications of a modification are not readily determinable for most systems, re–evaluation and recertification necessitated by maintenance may be a significant cost and risk factor for both developers and evaluators. Even a minor change to a system that involves a life–critical function has the potential for dangerous consequences without careful analyses and tests to assure that integrity and safety are maintained.

The qualification of reusable assets may be affected by changes as well as the adherence to original trust properties. The implications of suggested modifications must be assessed carefully to determine the impact on asset reuse and system trust and performance. Modifications to the trusted portion of the system will, in all likelihood, require modification to the analytic materials that have been developed to assure the trust characteristics of the system. For example, in a TCSEC trusted system [19], a modification to the trusted computing base (TCB) will necessitate re–examination and possibly modification to the interpretation of the formal policy model and the covert channel analysis, as well as to the more directly related products, such as the design specification and the user documentation. Since for TCSEC products, the requirements for architectural constraints are so stringent, modifications introduce the risk of loss of evaluation rating. Even if the rating can be maintained, the cost and associated risk of re–evaluation is a non–trivial concern.

2.5 Ada as a Risk Mitigator

While the Ada language presents some near–term risks to development organizations lacking experience with its use, its primary role in the SCPM can be as a risk–reducer and a quality–enhancer. In particular, Ada may be able to provide a homogeneous representation for a system during several stages of its development. Ada can be used as a design language, allowing subsystem and module interfaces to be checked for consistency by the compiler early in the development process. Ada may be shown to be useful in the future as a formal specification language. This may reduce errors in the transition from specification to design to code. In addition, it may allow better synergism among automated tools used in these phases. Furthermore, use of Ada during multiple development phases may allow a uniform set of metrics to be applied to design, specification and code alike. Finally, Ada's packaging and generic constructs support and encourage software reuse. All of these potential benefits can contribute to reduced risk during development. As an example, the Ada Process Model [2] that has been developed at TRW provides a prescriptive Ada development approach tailored to MIL–STD 2167A which is more detailed but generally consistent with the overall framework of the SCPM described in this report.

2.6 Process Visibility and Control

The dynamic, risk–driven nature of the SCPM requires careful planning and effective management for project visibility and control. As emphasized in the original Boehm spiral model, planning is necessary for the overall project success and is an essential part of each spiral prior to transitioning.

A primary ingredient for control is the Risk Management Plan. This plan defines overall initial risk mitigation activities and identifies goals and decision points, as well as activity reviews and mechanisms for technical and management exchanges. The risk plan should contain a draft master schedule that permits the flexibility required for incorporation of the results of concurrent, overlapping, and phased risk mitigation spirals, reuse impacts and activities for specific project goals.

The plan should also allow for project visibility very early in the development process. Visibility is particularly important during requirements definition and the early design of trusted system elements, especially for a reuse–driven, high–risk trusted system development. Customers, independent reviewers, and contractor management need to have insight into both the project status and the major risk–based decisions that are being made. Participation in reviews and walkthroughs and documentation of engineering results and decisions remain important activities. An automated project process manager will provide valuable assistance in the planning, tracking and scheduling of the risk management process. Visibility into the project on the status and dependencies of project activities will be greatly facilitated within a SEE that provides computer aided process management to help implement the Risk Management Plan.

The SCPM risk plan should incorporate frequent informal technical and formal management reviews at major transition points. An open, visible project can encourage better communications between the technical personnel, management, customers, and evaluators, and can foster a mutual understanding of the goals and purposes of the reuse and risk management activities within the project. The focus of the multiple, concurrent activities of spiral–based development will be better grasped and the potential for misunderstandings reduced with a project emphasis on visibility and open communication.

A crucial task in controlling a spiral–based development process is the definition of criteria for completing one spiral cycle and initiating the next. This task is complicated by the fact that each spiral cycle may be defined dynamically, making it necessary for each cycle to define the criteria for its own completion. The completion criteria must also be consistent with overall cost, schedule, staffing, and other parameters, and with progress assessment and review procedures established in the risk plan. It is essential that each cycle's criteria provide a clear cut–off point for termination of the cycle in order to discourage iterative reconsideration of alternatives and re–assessment of risks past the point of diminishing returns. Possible transition criteria for a cycle include elimination of risk for a particular risk item, delivery of documents or attainment of a consensus by a working group. In some cases, further control over the process may be achievable by refining the cycle transition criteria into transition criteria for individual quadrants within a cycle. Process control requires a forcing function to dictate decisions, enforce transitions and guide the overall project toward successful completion.

Automated SEE support for process management will be a necessary ingredient for success in the effective application of the process model. Both reuse and trust require stringent configuration controls to ensure strong confidence in the integrity of the assets and the overall environment that supports the development process. Tracking progress throughout the complex, concurrent risk mitigation activities of early spirals will be exceedingly difficult without automation. Project personnel must be well versed in the use and purpose of the SEE tools that apply and be deeply involved in the planning processes that motivate the use of the applicable tools.

3 SUMMARY AND CONCLUSIONS

This report documents an enhanced, spiral-based, life–cycle process model for the development of trusted, high performance systems within a reuse–based software engineering environment. Building on previous process modeling research, the SCPM described here is aimed at providing a process model description for trusted systems applicable to the STARS Tier 2 process, reuse and SEE goals. Overall paradigm objectives are to provide a top levei basis to support the identification and creation of reuse process building blocks, to support the analyses and specification for process management automation and to assist with the identification and planning for tool integration, adaptation and tailoring within an interoperable domain–specific SEE. The SCPM will be further tailored to include the domain–specific results of this TRW US40.2 task. Based on a documented set of Navy C^2 risks and characteristics identified through domain analysis, the final process model will provide a more specific description of process activities.

The SCPM defines five conceptual spirals of activities that address the significant risks in each of the life–cycle stages in the trusted system development process. Within a particular major spiral, there may exist subspirals of activity that address specific risk mitigation strategies. The spirals may be concurrent and overlapping or may be phased. Conceptualizing and tracking the potentially complex activities, especially early in the project when risks are the highest, are challenging tasks. This report emphasizes the need for project visibility and control supported by process automation within an integrated development environment. Proposed activities to support risk mitigation and project goals are listed and discussed within each of the five major project stages:

1. Initial Project Plans and Analysis of Reuse, Trust and Performance Requirements;
2. Reuse and Trust Enforcement Strategy and Basic Architecture;
3. Critical Elements and Architecture Refinement;
4. System Development and Assurance;
5. Maintenance.

The SCPM needs to be described at lower levels of detail to be more prescriptive and to assist the determination of specific tools and development methodologies. Tailoring the paradigm to a particular application domain will guide a next level analysis that provides more specific process details. The Navy C^2 domain analyses will support the elaboration of the reuse process and provide insight into more detailed technical and programmatic risks and activities associated with building reuse–driven, trusted Navy C^2 systems. However, even this more detailed set of activities will not provide a "cookbook" for its application. Once a lower level process description for trusted Navy C^2 system development is defined, the process model will need automation and validation through real–world applications. In addition, open research questions and major challenges remain to be resolved before the realization of STARS goals for software development improvements through advances in process definitions, reuse capabilities and the availability of SEE support.

3.1 Open Issues

Reuse methodologies require well-defined processes and useful metrics (including reuse criteria) for cost analyses and for reuse asset qualification/certification that are compatible with metrics in current use and evolving national and international standards. Successful reuse applications will depend on the quality of domain assets and the support environments that assist the reusers. Much experimentation with reuse applications and capturing of domain knowledge in several areas will be necessary before effective tools can be derived and reuse engineering can become cost effective and widely used.

There is no well defined, single technology for the achievement of broad trust once the clearly-bounded, confidentiality-based trust areas are expanded to include integrity, assured service, safety, high reliability, mission criticality and functional correctness. Research is needed in the community external to STARS in the specification of trust policies for systems whose trust is broader than confidentiality. Research is also needed (even in the area of confidentiality) to stretch current technology for engineering and developing trusted distributed systems and heterogeneous interfaces, for reusing and maintaining trusted elements and for evaluating trusted components and systems.

No definitive guidance exists for the domain analysis process in support of reuse technology. Domain analysis requires the input of experts and is a complex and difficult activity to perform. While domain analysis is perceived to be the first step in constructing reusable resources, there is no large body of knowledge or literature on the process or the products [6].

Software development process automation and configuration control for reuse and trust are necessary areas of further work to support the near term and long term goals for productivity advances in software development. More work is needed in support tools and configuration management and control to provide for management of concurrent activities within the SCPM. The evolution of a SEE that can support a risk-driven process model will require automation of a spiral-based process manager. The flexibility inherent in a spiral-based model which allows concurrent activities and other lifecycle options, can be more difficult to control than traditional, rigidly staged models. Process management tools will be needed to support the evaluation of spiral transition criteria, to assess project process, to enforce decision making, and to achieve consistency of notation for project management and reuse capabilities.

Research into appropriate formal methods is needed as well as more guidance on the appropriate role of formal methods in the development process and in the reuse process. The reasoning-based approach of the SCPM provides a framework for the use of formal and rigorously-applied informal techniques to the system development process. A rigorous approach provides a more sound, mathematical foundation for making assurance claims about the trustworthiness of systems and components under evaluation. Since the types of systems that require broader notions of trust are often very large, formal methods must be selectively applied in a meaningful way. The reuse of trusted assets and their associated formal assurance is a desirable cost saving goal. More research is needed for analysis of the portability oi trusted components and products to different applications within the same domain and in other application domains. It is expected that some of this research will be carried out in the broader based software engineering and computer security communities.

Architecting reusable, trusted systems will require further attention into the integration of reusable components with new components, the partitioning of a system into trusted and untrusted components and domain engineering for a specific application domain. Methodologies for defining and/or reusing the architecture for a trusted system must address the feasibility of isolating the critical components (functions that implement trust policy). Domain analysis technology must address the feasibility of generic architectures that can be reused and the tailoring that will be required for a new or updated system.

The achievement of SEEs to enhance software development productivity is a major STARS' goal for the 1990's. In particular, the environmental support for development of highly trusted systems under a reuse-based methodology will require additional capabilities. For secure developments, the overall considerations include computer, communications, personnel, physical, environmental, and procedural security requirements. The support tools used for system development, configuration management, asset qualification, and certification support must be trusted to perform as intended. Trusted tools to support trusted system developments are not currently available. Present practices do not ensure that the trustworthiness of the environment will match the required trust for the system under development.

3.2 Plans for Tailoring the Process Model to the Navy C^2 Domain

The initial tailoring of the STARS risk-reduction, reasoning-based Composite Paradigm to the trusted Navy C^2 environment is a major goal for this subtask. Government, TRW and Unisys domain experts will support the TRW process model team with a preliminary characterization of the Navy C^2 domain. For this task, TRW is also identifying the risks associated with the development of trusted Navy C^2 systems. Information derived from this analysis will help to define process model techniques and transitioning criteria for assessing and resolving the major risks in the trusted Navy C^2 domain.

The SCPM will be tailored to describe a process model for building trusted Navy C^2 systems that includes the application of domain and reuse technology. The process may include descriptions of Navy C^2 domain analysis activities as an early spiral that precedes the requirements analysis spiral. The domain-based process model description will be more explicit than the SCPM and will specify steps for activities and identify tighter dependencies. In addition, the model will address more specifically project deliverables and their production and management as necessary. The definition of lower level project activities also will help to identify automated tools that can be used to support the development process. The implications of Navy C^2 architecture(s), trust and reuse goals, evolvability needs, high performance, human factors and automated support requirements will collectively influence the process model descriptions.

Applications of the SCPM to Navy C^2 developments will help to validate its usefulness and appropriateness. In addition, the process descriptions are expected to be useful in support of the development of reuse process building blocks and in determining appropriate process automation support needed in a SEE. Selected building blocks could then be validated on a Navy C^2 development or in a shadow experiment to determine the viability of the risk-driven process and/or compare it to the more standard waterfall method in practice today.

References

[1] Boehm, Barry W., " Spiral Model of Software Development Enhancement," *IEEE Computer Surveys*: 61–72, May 1988.

[2] Royce, Walker W. "Incremental Development of Large Ada Systems: An Ada Process Model," *Proceedings of the 1989 ACM Tri-Ada Conference*, October 1989.

[3] Software Productivity Solutions, Inc., *Impact of Domain Analysis on Reuse Methods*, prepared for U.S. CECOM Army Center for Software Engineering, November 1989.

[4] *STARS UR40 - Repository Integration: Review of Existing Repository Technology*, Unisys Defense Systems, Paoli, PA, February 1989.

[5] *Framework of Issues in the Reuse of Trusted Software*, Unisys Defense Systems and Trusted Information Systems, July 1990.

[6] Holibaugh, Robert et. al., "Reuse: Where to Begin and Why" *Proceedings of Tri Ada 1989*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, September 1989.

[7] TRW Systems Division, *Process Model for High Performance Systems in Ada, Phase I Technical Report*, Fairfax, VA, August 1989.

[8] Shu, Christine, *Experience with Using V base and APPL/A for Process Modeling and Programming*, TRW Arcadia Project, January 1990.

[9] Kitaoka, Beverly J. SAIC, "Repository Support for a Reuse Process," *Proceedings of the 8th National Conference on Ada Technology*, Atlanta, GA, March 1990.

[10] Sutton, Stanley M., et.al., *Language Constructs for Managing Change in Software Process Programs*, University of California, Irvine, CA. and University of Colorado, Boulder, CO, August 1989.

[11] Creps, Richard "Unisys STARS Reuse Technology," Unisys Defense Systems briefing, September 1990.

[12] Solderitsch, James J., et.al., "Constructing Domain-Specific Ada Reuse Libraries," *Proceedings of the 7th Annual National Conference on Ada Technology*, March 1989.

[13] Balzer, Robert, et.al., *Software Technology in the 1990's: Using a New Paradigm*. November 1983.

[14] McCracken, Daniel and Jackson, Michael, "Life-Cycle Concept Considered Harmful," *ACM Software Engineering Notes*: 29–32, April 1982.

[15] Royce, Walker W.. TRW Systems Engineering & Development Division, *Ada Process Model*, November 1989.

[16] Solderitsch, James and Payton, Teri, Unisys Defense Systems, *A Basis for Domain Specific Support Environments*, May 1990.

[17] *UR 40: Repository Integration: Draft Definitized Repository Specification*, STARS-RC-01240/001/00, Unisys Defense Systems, Paoli, PA, July 1991.

[18] Layman, Gene, "Ada Repository Program," Naval Research Lab Briefing, October 1990.

[19] National Computer Security Center, Department of Defense, *Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD, December 1985.

[20] *SDI-BMS Security Accreditation Study*, Phase 1 Report, Technical Report TM-(L)-8361/004/00, Unisys Corp., Camarillo, CA, March 1988.